A parallelized Python-based Multi-Point Thomson Scattering analysis in NSTX-U

Jared Miller[1], Ahmed Diallo[2], Benoit LeBlanc[2]

[1]Northeastern University, [2]Princeton Plasma Physics Laboratory

Motivation

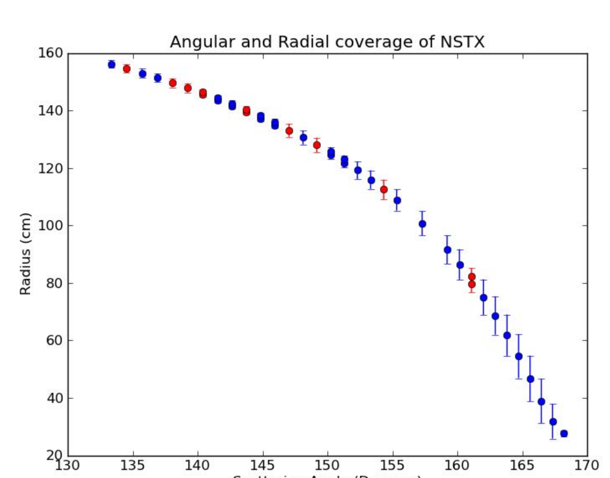
Multi-Point Thomson Scattering (MPTS) is a reliable and accurate method of finding the temperature, density, and pressure of a magnetically confined plasma at multiple positions over time. The existing code to perform an MPTS analysis for the National Spherical Torus Experiment (NSTX) was written serially in IDL and has accumulated since 2000. This project replicates and extends the functionality of the original IDL code by porting it to an object-oriented and parallelized Python implementation with refactored and further optimized computations, all for use in the NSTX upgrade (NSTX-U).

Background

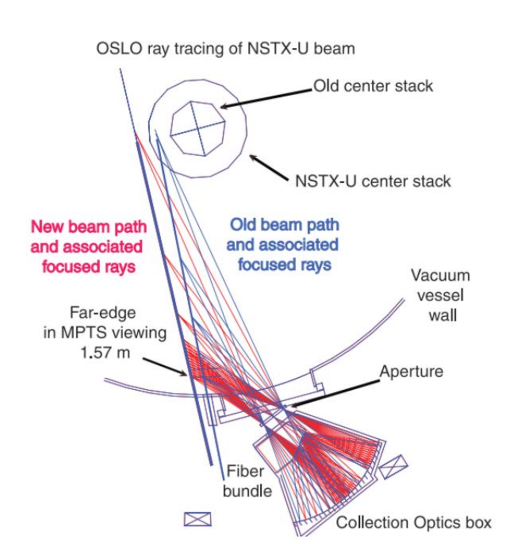
MPTS is an extraordinarily useful diagnostic used in reporting the behavior of plasmas. In this method, lasers are fired into the plasma, and the resultant light spectrum released by the plasma is picked up by detectors. The Doppler shifting of the supplied light due to Raman scattering is linked to the electron temperature, and the intensity of the light is proportional to the electron density. The resultant pressure is proportional to the density*temperature.

Two Nd:YAG (1064.1 nm) lasers alternately fire light into the plasma at 30 Hz, providing an equivalent rate of 60 Hz. When the photons interact with the electrons of the plasma, the light can be reemitted while keeping the same energy (Rayleigh scattering), or the energy of the light can change due to the electron either gaining or losing energy (Stokes or anti-Stokes Raman scattering). Fiber optic bundles are placed around the midplane of the tokomak to channel the emitted light into arrays of detectors. At each radial position where there is a fiber bundle probe, the light is sent into a polychromator containing either 4 or 6 channels. Each channel has a bandpass filter which only lets light of a specific wavelength range through to an avalanche photodiode (APD). Each APD outputs a voltage difference which is proportional to the number of photons. For each time step, the APD capture a slow signal (long sample time) and a fast signal (short sample time). The fast signal includes the laser pulse and contains the data, while the slow signal captures background noise and is used to find the error in measurements. In total, the raw signal output of all of the detectors is a voltage for each target wavelength per radial position around the tokomak.

The polychromators (otherwise referred to as shelves) are grouped ac cording when they were installed. Phases 1 and 2 have 10 shelves each, and they each have 6 channels (1 Rayleigh, 5 Raman) and phase 3 has 10 shelves with 4 channels (4 Raman). Phase 4 is being installed as part of the NSTX upgrade project, and is composed of 12 shelves with 6 channels (1 Rayleigh, 5 Raman like phases 1 and 2).



The blue points are shelves in phases 1, 2, or 3. The red points are shelves in phase 4With the addition of phase 4 shelves, about 90% of the machine apeture is covered.



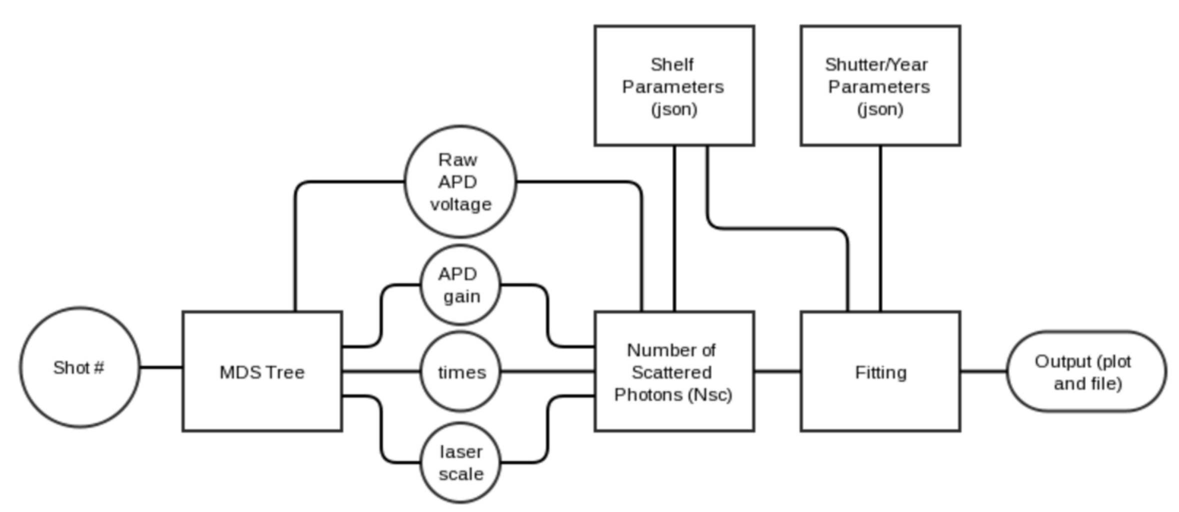
The old beam path would have collided with the newer wide center stack. (OSLO render)

Tools

- Python 2.7.2 (nstx/python-2.7)
- MDSPlus (mdsplus5, the tree database which stores all shot data)
- Numpy (native numerical computations)
- Scipy (advanced numerical algorithms)
- Matplotlib (plotting and display)
- Multiprocessing (parallelization)
- JSON (human readable configuration input)

Pipeline

This sequence of steps is performed once per polychromator. The current Python implementation of MPTS parallelizes the pipeline using the Multiprocessing module. This process is run across all 30 or 42 shelves in parallel.

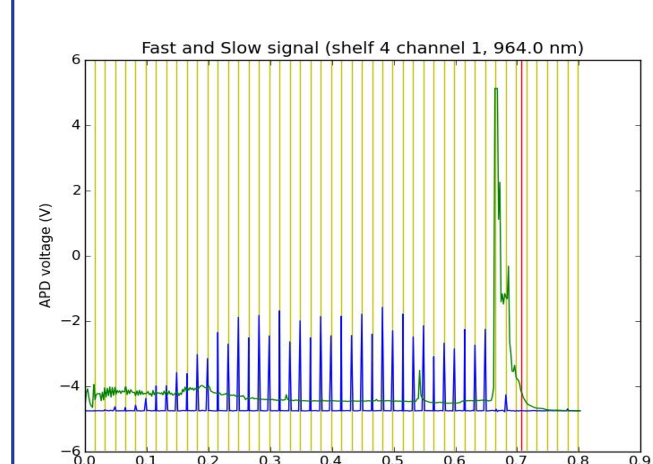


Calculating Gain

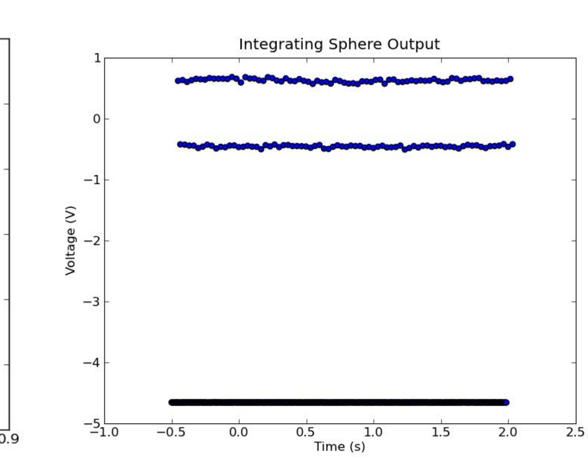
The gain of each APD is derived from two sources, the electronic gain and the physical gain. The electronic fast signal gain is always set to 10, while the slow signal gain is either 20 or 40 depending on the wavelength. The physical gain is calculated by reading the applied voltage values from the MDS tree (VDAC) and finding the roots of a cubic equation with the coefficients given in the per shelf calibration. The coefficients also depend on whether the APD's are in linear mode or temperature mode. The exponential of each root is taken, and the purely real result closest to 70 is the physical gain.

Timing

The only times where there are valid data are when one of the lasers fire. In order for a time point to be flagged for further evaluation, the fast signal must have a peak. As an error checking mechanism, a beam splitter diverts some of the laser energy to an integrating sphere which outputs a voltage. When both the integrating sphere and the fast signal have a peak and also the plasma current is greater than zero, then that time point is valid.



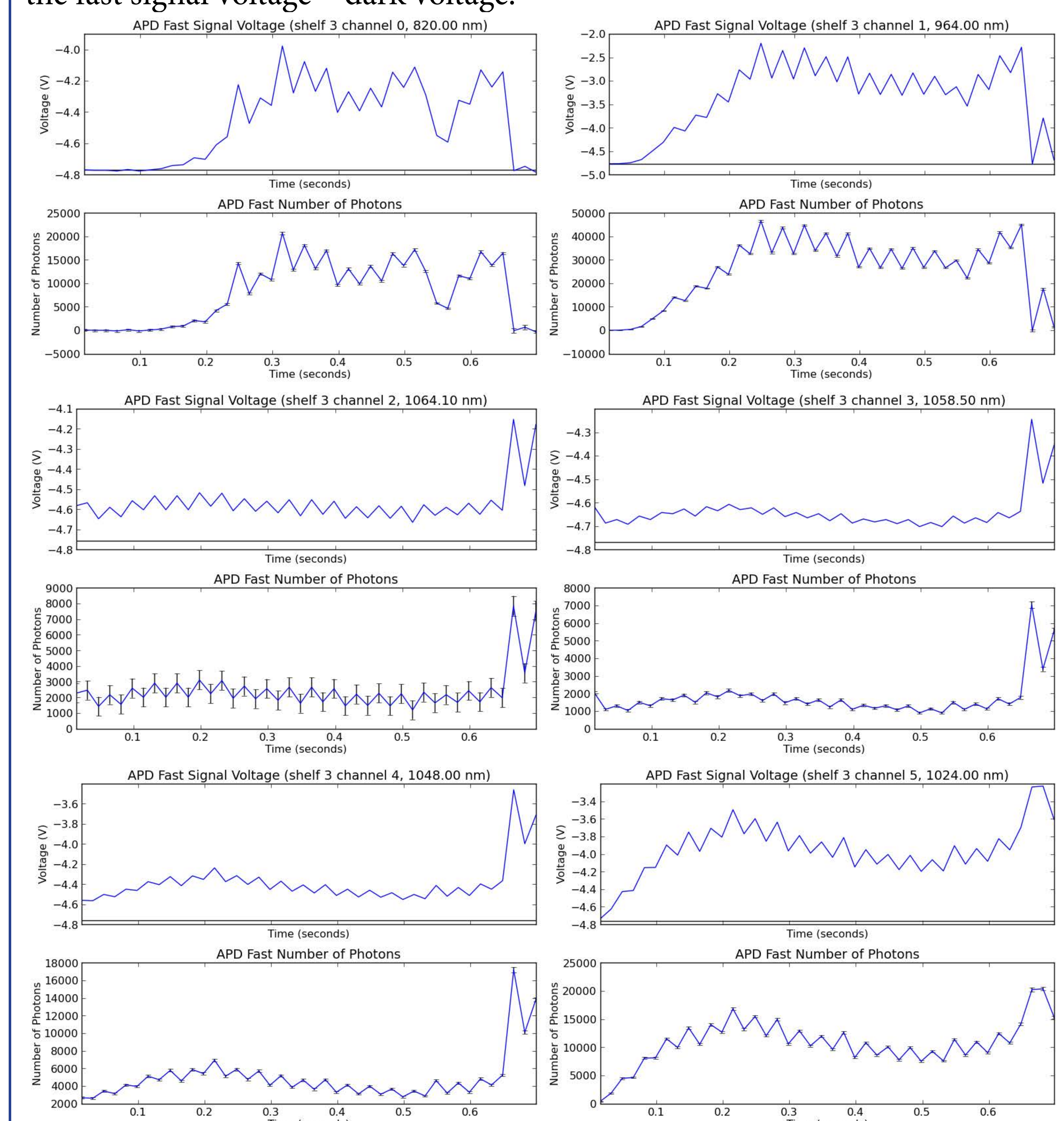
Fast (blue) and slow (green) signals. Yellow lines are laser pulses, which line up with fast signal peaks. Red line is when plasma current goes to zero. Data is from shot 130000.



Output of the Integrating Sphere. Time points are clearly grouped into Laser 1 pulse (0.7 V), Laser 2 pulse (-0.4 V) and no laser.

Number of Scattered Photons

The Number of Scattered Photons (Nsc) that an APD detects is proportional to its voltage difference. The voltages before t=-0.25 are averaged to make a baseline dark voltage. At every relevant time point, the voltage difference is the fast signal voltage – dark voltage.



This plot shows the raw voltage and number of scattered photons (Nsc) for each channel of shelf #3 (68.64±6.67 cm). Channel 2 is for Rayleigh scattering, as it only detects photons of the laser wavelength, 1064 nm. Rayleigh scattering does not occur until the plasma starts to dissipate. The other channels detect Raman (non-laser) wavelengths. The black line in the top plots is the baseline (dark) voltage.

Fitting

A.C. Selden introduced a formula in 1980 which calculates the number of scattered photons on an APD given the electron temperature (Te), electron density (ne), and a series of other parameters. All of the other parameters are stored inside the configuration files, so Te and ne must be solved for. At each relevant time point for each polychromator, there are 4 or 5 values of Nsc, since the Rayleigh APD is excluded. A weighted, nonlinear, least squares fit with χ-squared minimization is used in order to solve for ne and Te. Scipy's curve_fit() routine uses the Levenberg–Marquardt algorithm in order to find this solution of solving 4 or 5 equations in 2 unknowns. If an APD has an Nsc below a threshold (such as -40 photons) and the Nsc + error < 0, then that APD is excluded from the fit. After this pruning, if 2 or fewer equations must be solved in 2 unknowns, the value of ne and Te for that data point is interpolated later.

$$N_{sc} = \left(\frac{n_e r_e^2 E_L L\pi}{4 f^2 hc}\right) \int C(\alpha) \frac{(1+x)}{A(x,\theta)} e^{-2\alpha B(x,\theta)} Q T_p T_c \, d\lambda$$

where,

$$\alpha = \frac{m_e c^2}{2kT_e}, \quad x = \frac{\lambda - \lambda_i}{\lambda_i},$$

$$A = (1+x)^3 \left[2(1-\cos\theta)(1+x) + x^2\right]^{1/2},$$

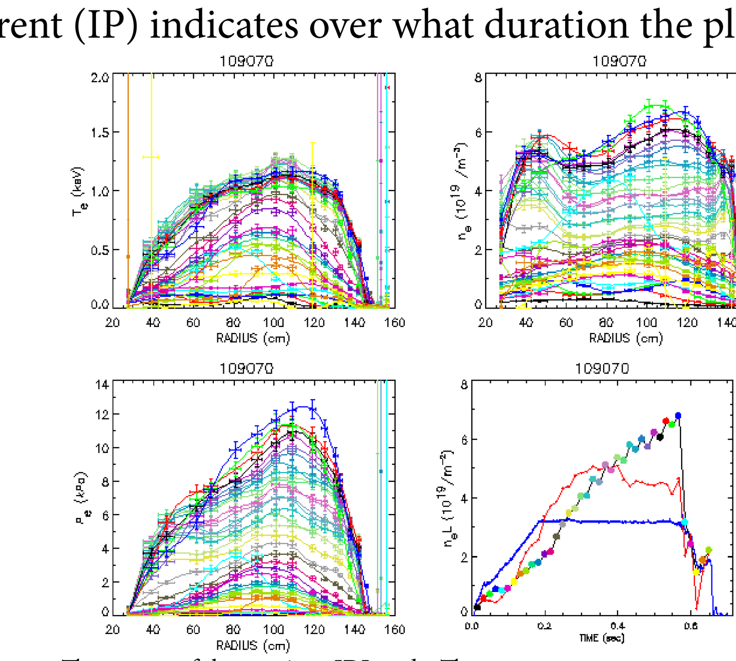
$$B = \left[1 + \frac{x^2}{2(1-\cos\theta)(1+x)}\right]^{1/2} - 1,$$

$$C = \sqrt{\frac{\alpha}{\pi}} \left(1 - \frac{15}{16}\alpha^{-1} + \frac{345}{512}\alpha^{-2}\right).$$

The Selden formula, predicts number of scattered photons given density (n_e) and temperature (T_e)

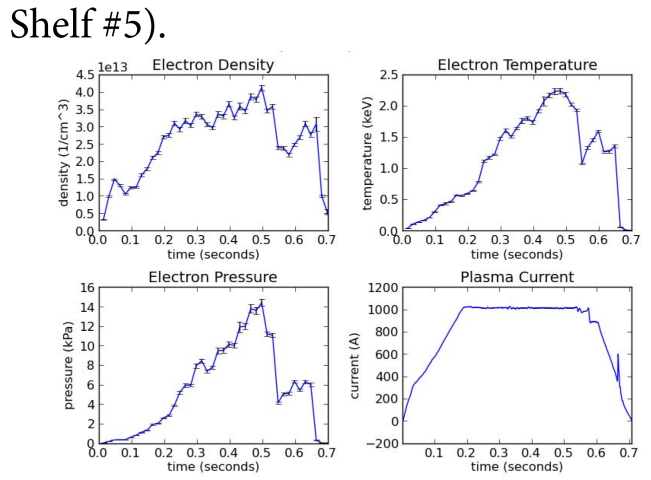
Output

The output is a 4 part plot which displays the temperature, pressure, and density over time with associated error bars corresponding to one standard deviation. The lower right subplot is a plot of the plasma current going around the tokomak, and is the same for all polychromators. The plasma current (IP) indicates over what duration the plasma exists.



The output of the previous IDL code. The pressure, temperature, and density of the 30 shelves are presented in this output plot. (Figure from 2003, B.P. Leblanc)

Currently, the code does not yet match IDL. Below is an example of what the output would look like (note that the pressure, density, and temperature are incorrect, Shelf #5).



Future Plans

The main goal for the near future is to debug and make sure that the python code matches the IDL code. Once identical results are reliably produced, the Python code can replace the IDL code for use in the NSTX-U. The 4-port plot can also be composed of 3d plots in order to fully visualize the plasma across both radius and time. Once the Python code is given MDS writing access on top of reading access, the program can add the results of the computation to a shot's tree for future use. The user interface can also be improved, and a GUI can be constructed instead of strictly using the command line. Once all of these tasks are completed, the code can be adapted for any polychromator based MPTS system.

Further Information

Jared Miller	miller.jare@husky.neu.edu
Ahmed Diallo	adiallo@pppl.gov
Benoit LeBlanc	leblanc@pppl.gov

PRINCETON PLASMA PHYSICS LABORATORY